Within 5 Years – Most Mobile PCs will have Tablet PC Functionality”

– *Bill Gates*

MBL343
Getting Started With The Tablet PC Platform

David Hale
Lead Programmer/Writer
Tablet PC Developer Experience Team

Microsoft Corporation

Microsoft Tech·Ed 2004

Agenda

- Tablet PC Platform Overview
- Description of the Object Model
- New Features in version 1.7 of the Tablet PC Development Kit
- Development
- Deployment

Tablet PC Platform History

	V1.0/1.1	V1.5	V1.7
API:	Ink	Same As V1	Same As V1
	Collection		StylusInput APIs
			Web Support
	Ink		
	Data		
	Ink	Same As V1	Same As V1
	Reco		Web Support
Controls:	InkEdit	Updated Docs	Updated Docs
	InkPicture	Updated Samples	Updated Samples
SDK:	Docs		
	Samples	PenInputPanel	Same As 1.5
		InkDivider	
Components:	n/a		

Tablet PC Platform Overview

- Ink Collection (Input)
- Ink Data and Management
- Ink Recognition

Ink Collection (Input)

- Digitizer Basics
  - Very High Resolution and Throughput
  - Packet Properties
    - X, Y
    - Pressure, Angle, Rotation, etc …
- Using the API to collect Ink
  - InkCollector
  - InkOverlay
  - RealTimeStylus (New in SDK version 1.7)

Ink Data and Management

- Ink is a data type
- Multiple persistence formats
- Ink looks good
  - Vector based
  - Bezier smoothed
  - Anti-aliased
  - Customizable rendering
- Rich, extensible API
- Copy to / paste from the Clipboard

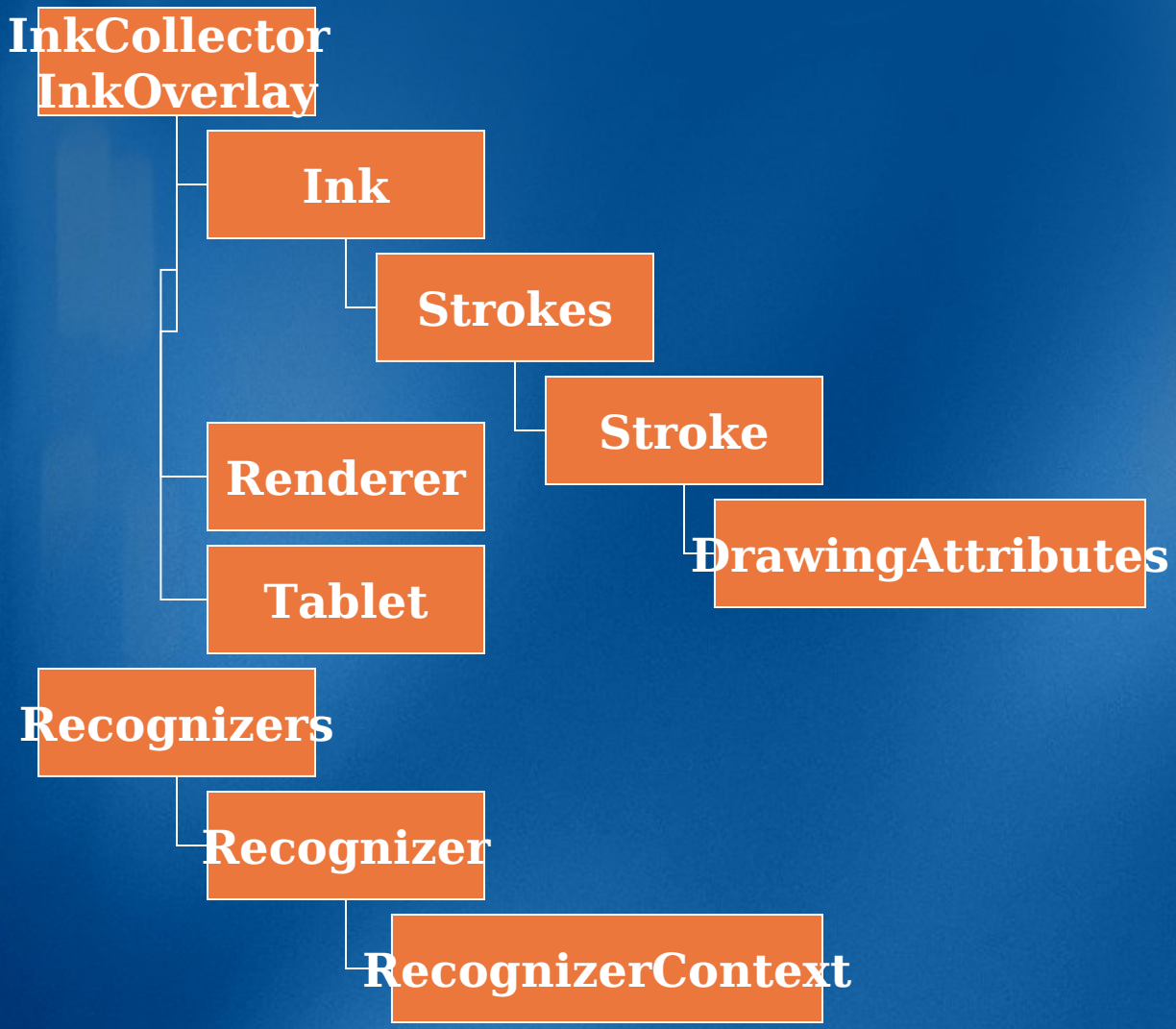
Ink Recognition

- Interpret pen movements and/or strokes as
  - Text (Handwriting)
  - Gestures
  - Shapes and symbols
- Language Support
- Synchronous and Asynchronous recognition supported
- Provides a full set of recognition results
- Extensibility architecture

Tablet PC Platform SDK

- 2 full APIs
  - COM Automation Compliant API
    - Dual Interfaced
    - IDispatch based callbacks for events
  - .NET API
    - Thin layer over the COM API
- Ink controls
  - InkEdit
  - InkPicture
- Documentation and Samples

Object Model Overview



Focus on Key Objects

- **Ink Collection**
  - **InkCollector**
  - **InkOverlay**
  - **Ink Collection Events**
- **Ink Data Management**
  - **Ink**
  - **Stroke**
  - **Strokes**
  - **DrawingAttributes**
  - **Renderer**
  - **Ink Data Events**
- **Ink Recognition**
  - **RecognizerContext**
  - **RecognitionResult**

Ink Collection Objects

InkCollector

- Top level object for collecting ink
- Attaches to any HWND
- Handles input from all attached Tablets (digitizers) and Cursors (pens)
- Intercepts cursor movements
- CollectionMode property
  - Ink Only
  - Gesture Only
  - Ink and Gesture

InkOverlay

- Superset of InkCollector
- Adds Editing Modes
  - Ink – Real-time inking mode
  - Select – Selection mode
  - Delete – Eraser mode (both point and stroke)

Ink Collection Events

- CursorInRange/CursorOutOfRange
- NewInAirPackets
- CursorDown
- NewPackets
- Stroke
- Gesture
- SystemGesture
- Mouse Events

demo

Ink Collection

David Hale
SDK Lead
Tablet PC Developer Experience Team

Ink Data Objects

Ink Object

- Container for
  - Stroke Data
  - Metadata
- Controls all means of persistence
- Can be extended using the ExtendedProperties property

Stroke Object

- Represents a single stroke
  - Single pen down, move, pen up sequence
- ExtendedProperties
  - Allows application to store any data on the Stroke
- DrawingAttributes
  - Controls the rendering of the Stroke
- Rich API
  - PacketSize, PacketCount, PacketDescription

Strokes Collection

- Represents a collection of *references* to Stroke objects
- Implements ICollection, IEnumerable
- Group related Strokes together
  - For filtering - By metadata
  - For recognition – By setting the RecognizerContext.Strokes property

DrawingAttributes Object

- Controls the appearance of Ink
  - Color, width, transparency
  - Smoothing, anti-aliasing
  - Height, width
  - Pen Tip: Ball or Rectangle
- DefaultDrawingAttributes property
  - Specifies the DrawingAttributes for all Cursors
- Can be applied to
  - Stroke
  - Strokes collection
  - Cursor (Pen)

Renderer Object

- Two coordinate systems on a Tablet PC
  - Device coordinates
  - Ink coordinates (HIMETRIC units)
- Renderer object
  - Map between coordinate systems
  - Draw methods for rendering
  - Manipulation methods
    - Object and View transforms supported
    - Helper methods: Scale, Move, Rotate

Ink Data Events

- InkAdded, InkDeleted
  - Raised for Ink being added to or deleted from an ink collecting object
  - Raised for internal Strokes being added, like erasing

Clipboard Support

- Use the Ink API to copy and paste
  - ClipboardCopy
  - ClipboardPaste
- Multiple formats supported
  - ISF – Ink Serialized Format
  - HTML
  - Bitmap, Metafile
- Applications interacting with the clipboard can choose the format

demo

Ink Data Manipulation

David Hale
SDK Lead
Tablet PC Developer Experience Team

Ink Recognition Objects

RecognizerContext

- Is the instantiation of a given Recognizer
- Allows recognition of a collection of Strokes
  - Synchronous (foreground) recognition
  - Asynchronous (background) recognition
    - Raises events
      - Recognition
      - RecognitionWithAlternates
- Provides access to the recognizer result and alternates

RecognitionResult

- Recognizers contain a set of structures
  - Each structure called a “lattice”
  - Copied to RecognitionResult and others
- TopString property
- Store on the Strokes collection and add to Ink.CustomStrokes
  - SetResultOnStrokes method
- RecognitionAlternates object
  - GetAlternatesFromSelection method
  - Access to alternate paths through the lattice

demo

Ink Recognition

David Hale
SDK Lead
Tablet PC Developer Experience Team

Platform Controls & Components

Platform Controls

- InkEdit
  - Manipulate ink as text
  - Intended for text and ink-to-text scenarios
  - Enables smart forms
  - Superclass of the RichEdit control
  - Win32, .NET, ActiveX versions
- InkPicture
  - Ink-annotate images (.jpg, .bmp, .png, .gif)
  - Collect and save ink as ink
  - Subclass of PictureBox, akin to

Platform Components

- PenInputPanel
  - Programmatic access to the PIP (v 1.5)
    - Binds to any hWnd or Control
  - Basic control of the TIP (Lonestar)
    - Enable / Disable, etc.
    - Needed for backward compatibility
- Divider (InkDivider)
  - Divides Ink into words, lines, paragraphs and drawings
  - Transforms angled ink

New Features in the Version 1.7 Platform

- Context
- Web support
- New input features

Context

- What is context?
  - App-specific info you provide to help the recognizer
    - Constrains recognition rules/possibilities
    - Greatly improves recognition accuracy
- Examples of context information
  - Factoids: `recoContext.Factoid = “(!IS_DATE)”;`
  - Word and phrase lists
  - Custom input scopes with regular expressions
- Context-enabled applications in Lonestar
  - Internet Explorer (Address Bar)
  - Outlook 2003 (To:, Cc:, Bcc:)

Context for Non-Ink-Enabled Controls

- SetInputScope API (new WIN32 API)
  - Low-level programmatic solution
  - Binaries impacted
  - White paper available for .NET development
- Context Tagging Tool
  - Non-programmatic
  - Binaries not impacted
  - Persists input scope / field relationships with XML manifest

demo

Context Tagging

David Hale
SDK Lead
Tablet PC Developer Experience Team

Web Support

- Support for Partially Trusted callers
  - Host ink-enabled Windows Forms controls on Web pages
    - Ink on a Web page
    - Send ink to the Web server
    - Retrieve ink from the Web server
- No-touch deploy Tablet PC applications
  - http://www.Fabrikam.com/claims.exe
- Application examples
  - Inventory management
  - Expense reporting
  - Claims handling

Hosting Ink Controls

- Create the Windows Forms control
  - Ink-enable with InkOverlay object
  - Use InkEdit or InkPicture controls
- Deploy the control library to the Web server
- Create the Web page
  - <object> tag to hold the control
  - Script functions to invoke the control's exposed members
- More info: ASP.NET QuickStart Tutorial
  - "Using Windows Forms Controls in Internet Explorer"

StylusInput APIs

- Direct access to digitizer data stream
  - Manipulate packet data in real time
  - Separate Real Time Collection and UI threads
  - Better performance than InkOverlay, InkEdit, etc.
- Build custom components
  - Real Time Ink components
    - Custom dynamic rendering (e.g., hardware accelerated)
    - Custom gesture recognition
    - Disabled regions (WindowInputRectangle property)
  - Ink Collection components
    - Custom InkOverlay or InkCollector components

demo

RealTimeStylus

David Hale
SDK Lead
Tablet PC Developer Experience Team

Other Improvements

- Manipulation of ink
  - Create stroke with full packet description
- HTTP User-Agent
  - Now identifies Tablet PC OS and SDK version
- Enhanced SDK Documentation
  - Lots of new content, samples, etc…

Development Environment

- Install on non-tablet devices
  - SDK on Win2K SP3 and above
  - Windows XP Tablet PC Edition OS on non-tablet hardware
  - Microsoft Virtual PC
  - Available on MSDN
    - MSDN: SDK
    - MSDN Subscriber Downloads: OS
- Simulate tablet digitizer
  - External HID digitizer
  - Mouse (UseMouseForInput property)

Deploying your Tablet Application

- Redistributable merge modules
- Considerations
  - Recognition
  - Ink collection w/o a digitizer
  - PIP and TIP
  - No support for the Win9X platforms

announcing...

Tablet PC Developer Center

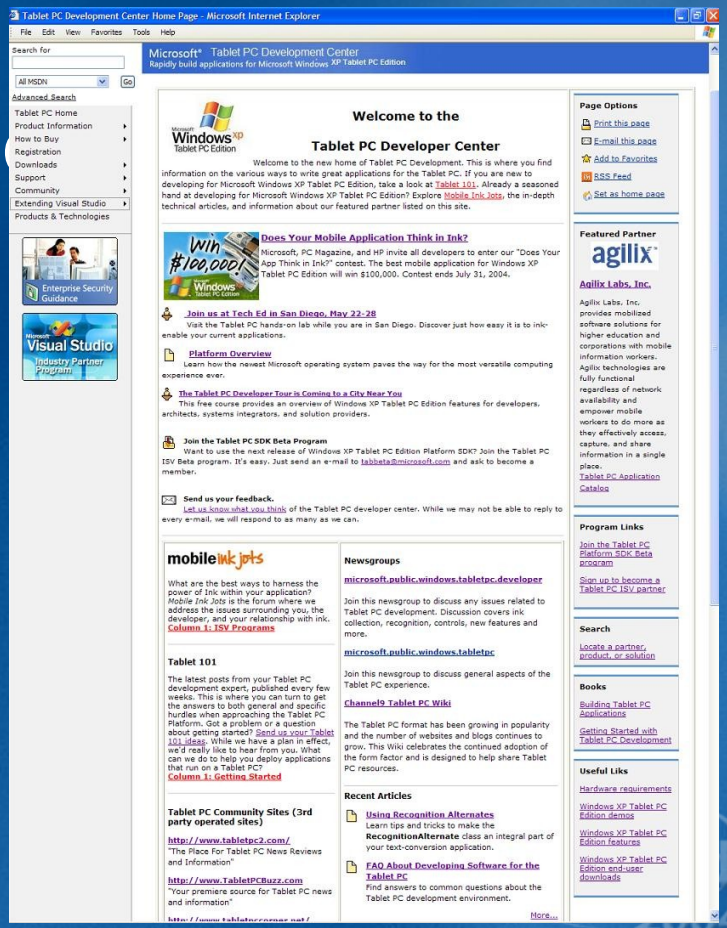
http://msdn.microsoft.com/tabletpc

Tablet PC Developer Center

- The latest technical articles
- Downloads
- Developer resourc
- Regular columns

mobile ink jots

- Newsgroups
- Featured partners
- and more……



Win $100,000!

DEVELOP THE **BEST APP** FOR THE EVOLUTION OF THE NOTEBOOK:

THE TRULY MOBILE TABLET PC.

Get ready
Microsoft® and PC Magazine® invite all ISVs to enter our "Does Your App Think in Ink? contest.
The best mobile application for Windows® XP Tablet PC Edition will win **$100,000**!
And you don't have to share any code with Microsoft.

Get set
You can develop your Tablet PC Platform SDK application on a standard Windows XP desktop.

Go!
Enter today. Contest ends soon.

www.doesyourappthinkinink.com

Windows xp Tablet PC Edition PC MAGAZINE hp invent Microsoft

Session Evaluation

Please fill out a session evaluation on CommNet

Q1: **Overall satisfaction with the session**

Q2: **Usefulness of the information**

Q3: **Presenter's knowledge of the subject**

Q4: **Presenter's presentation skills**

Q5: **Effectiveness of the presentation**

Questions?

Microsoft®

Your potential. Our passion.™